CMPT 733
Further Topics in Deep Learning
Sequence learning, Sentiment analysis, Word2Vec, DL-Vis

Steven Bergner

sbergner@sfu.ca

30 March 2020

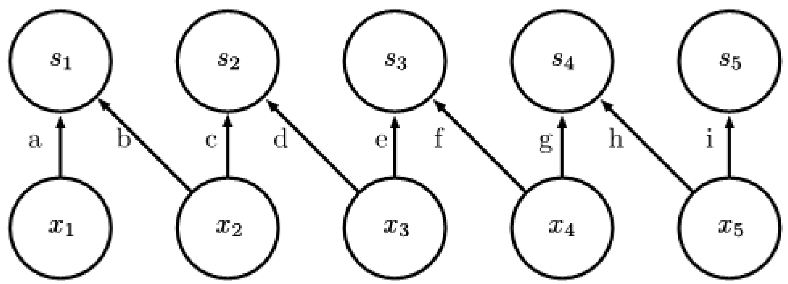
Overview

- Deep learning approaches for sequence learning with RNNs
- Natural language processing, e.g.
  - Sentiment analysis
  - Word embeddings
- Visualization for Deep Learning

Recap: Choosing architecture family

- No structure → fully connected
- Spatial structure → convolutional
  - Adjacency or order of inputs has meaning
- Sequential structure → recurrent

Types of connectivity

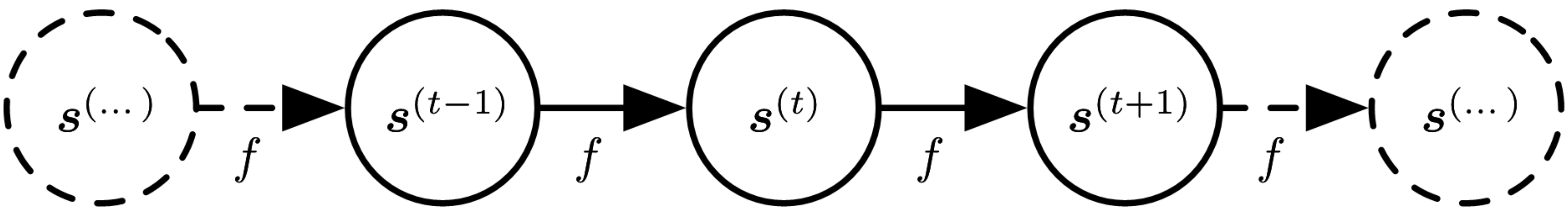


Local connection:
like convolution,
but no sharing

Sequence Modeling with Recurrent Nets

Classical Dynamical Systems

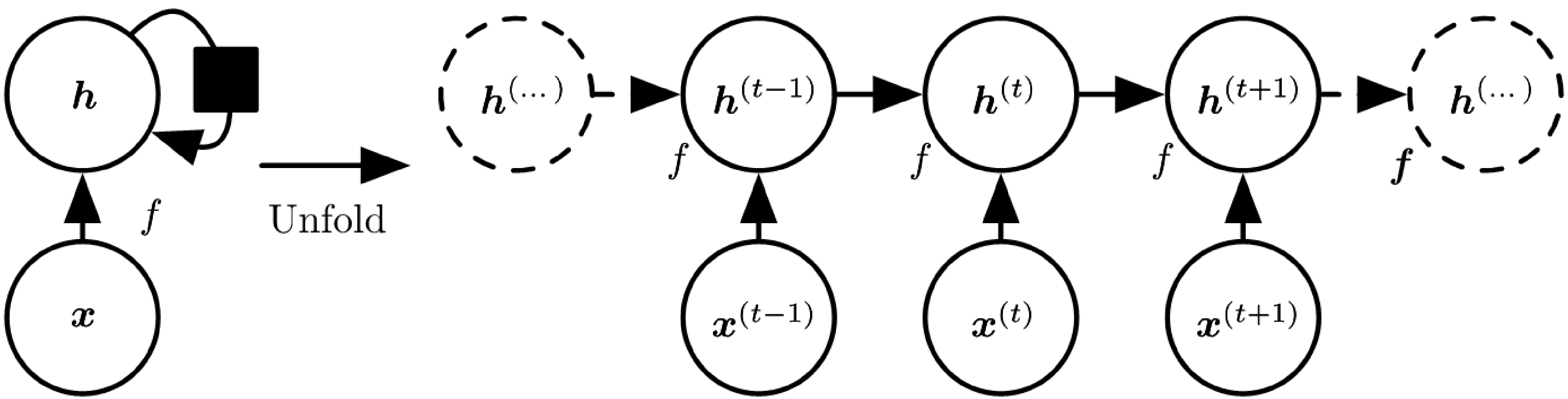
- Recurrent network models a dynamical system that is updated in discrete steps over time
- Function *f* takes input from time *t* to output at time *t+1*
- Rules persist across time



[Goodfellow, Bengio, Courville 2016]

Unfolding Computation Graphs

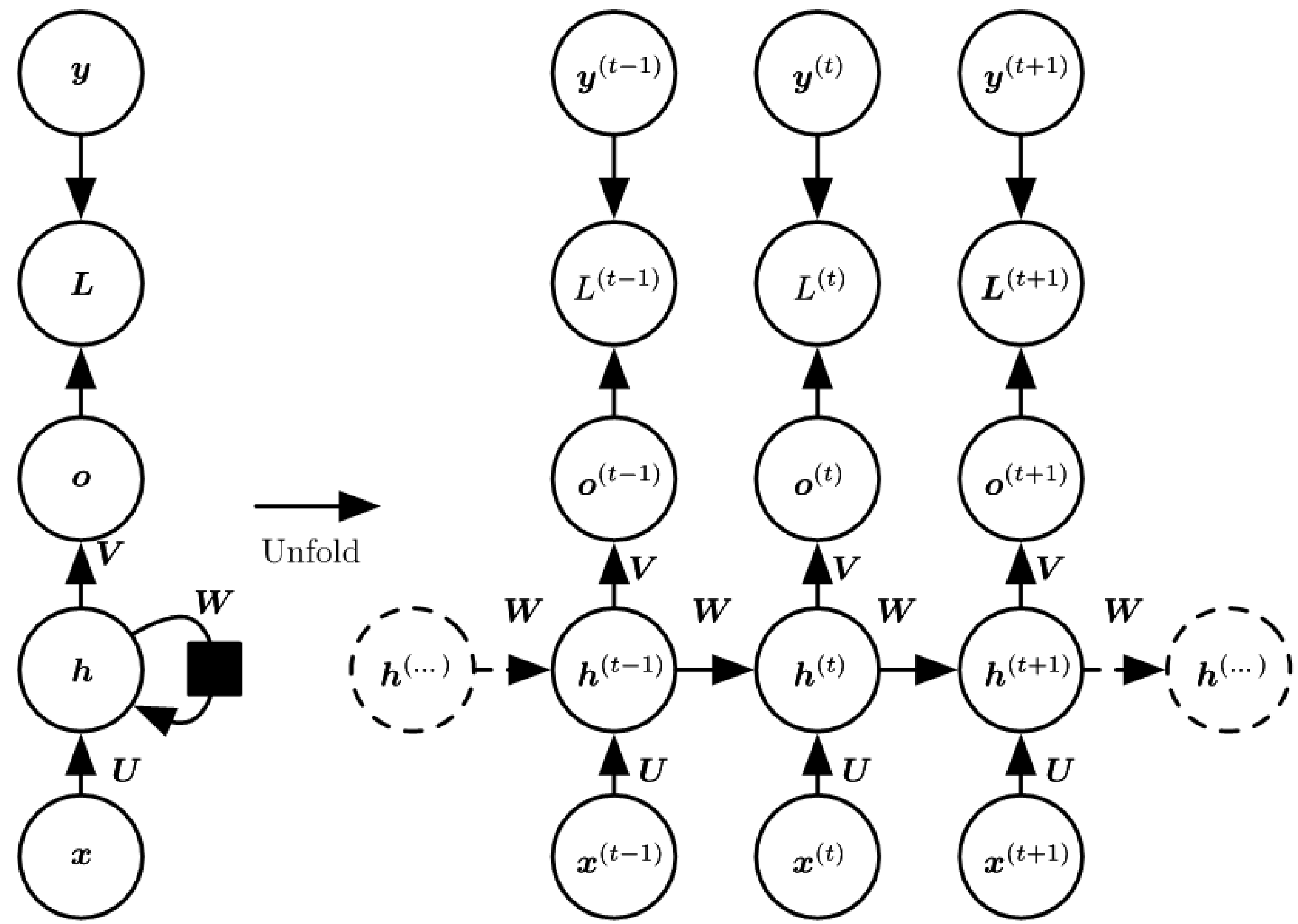
- Recurrent graph can be unfolded, where hidden state h is influencing itself
- Backprop through time is just backprop on unfolded graph



[Goodfellow, Bengio, Courville 2016]

Recurrent Hidden Units

- Can have more than one layer

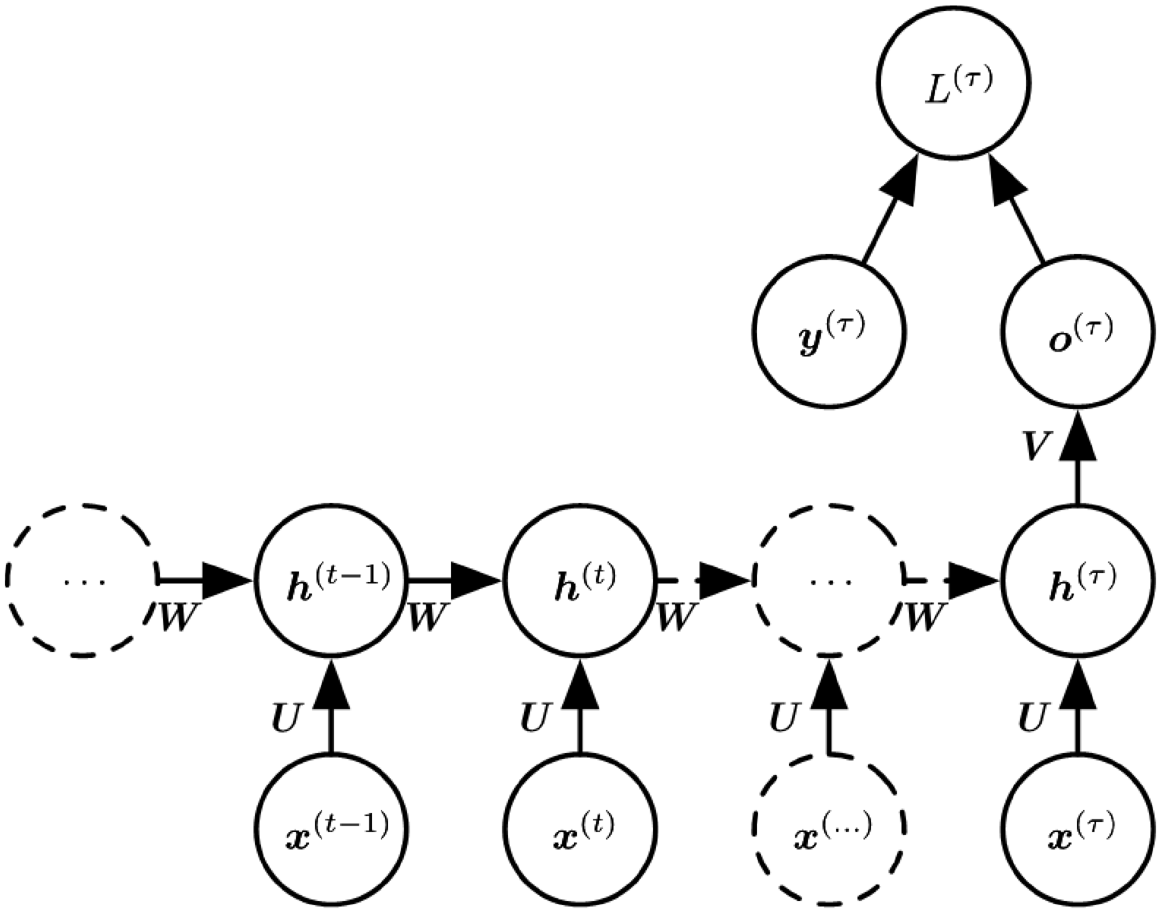


[Goodfellow, Bengio, Courville 2016]

Sequence Input, Single Output

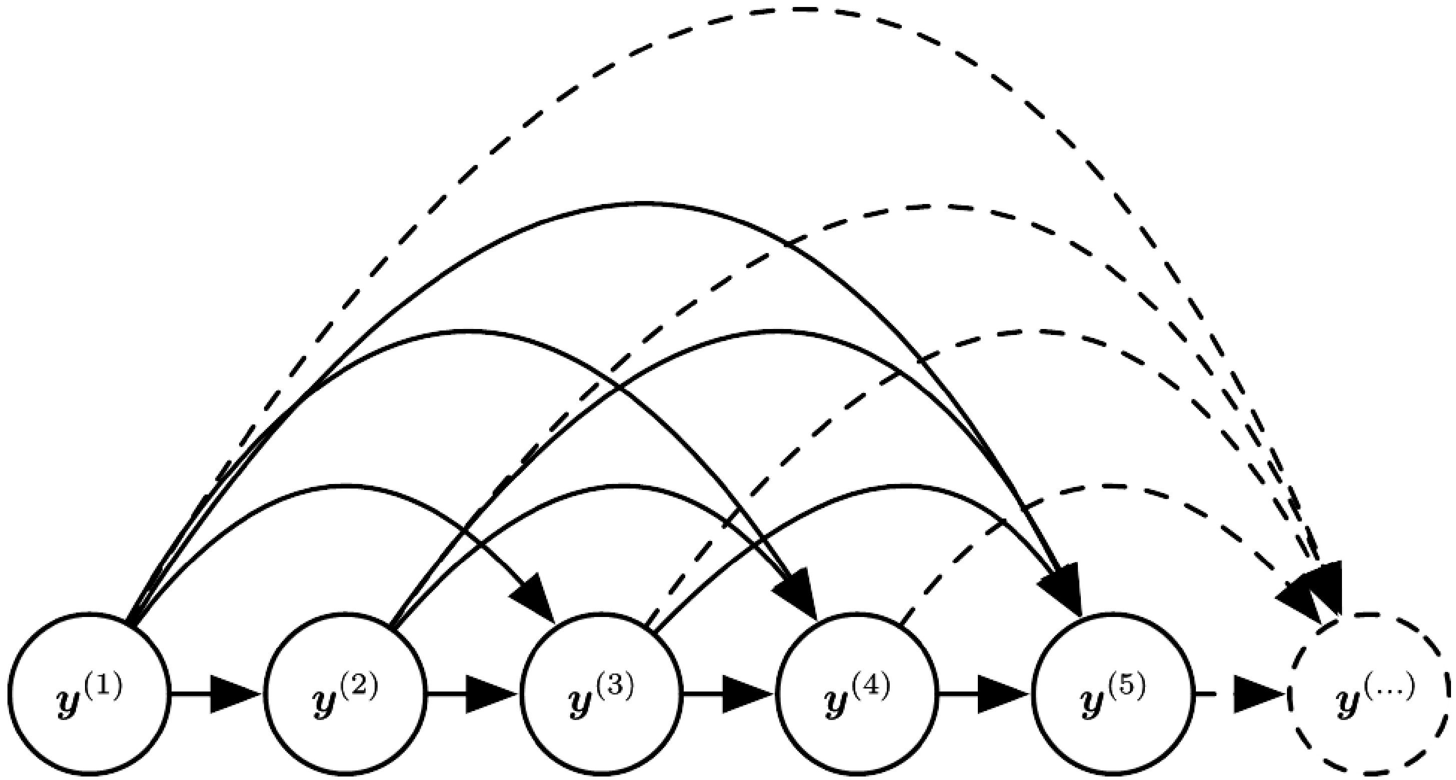
Example

Sentiment analysis of text



Fully Connected Graphical Model

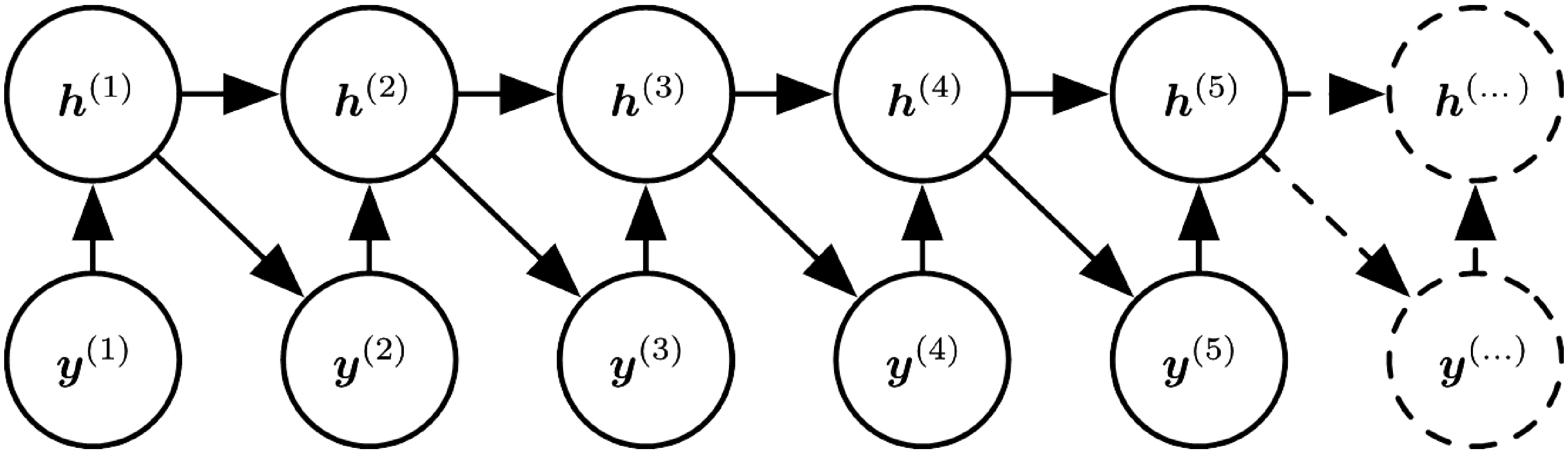
- Too many dependencies among variables, if each has its own set of parameters



[Goodfellow, Bengio, Courville 2016]

RNN Graphical Model

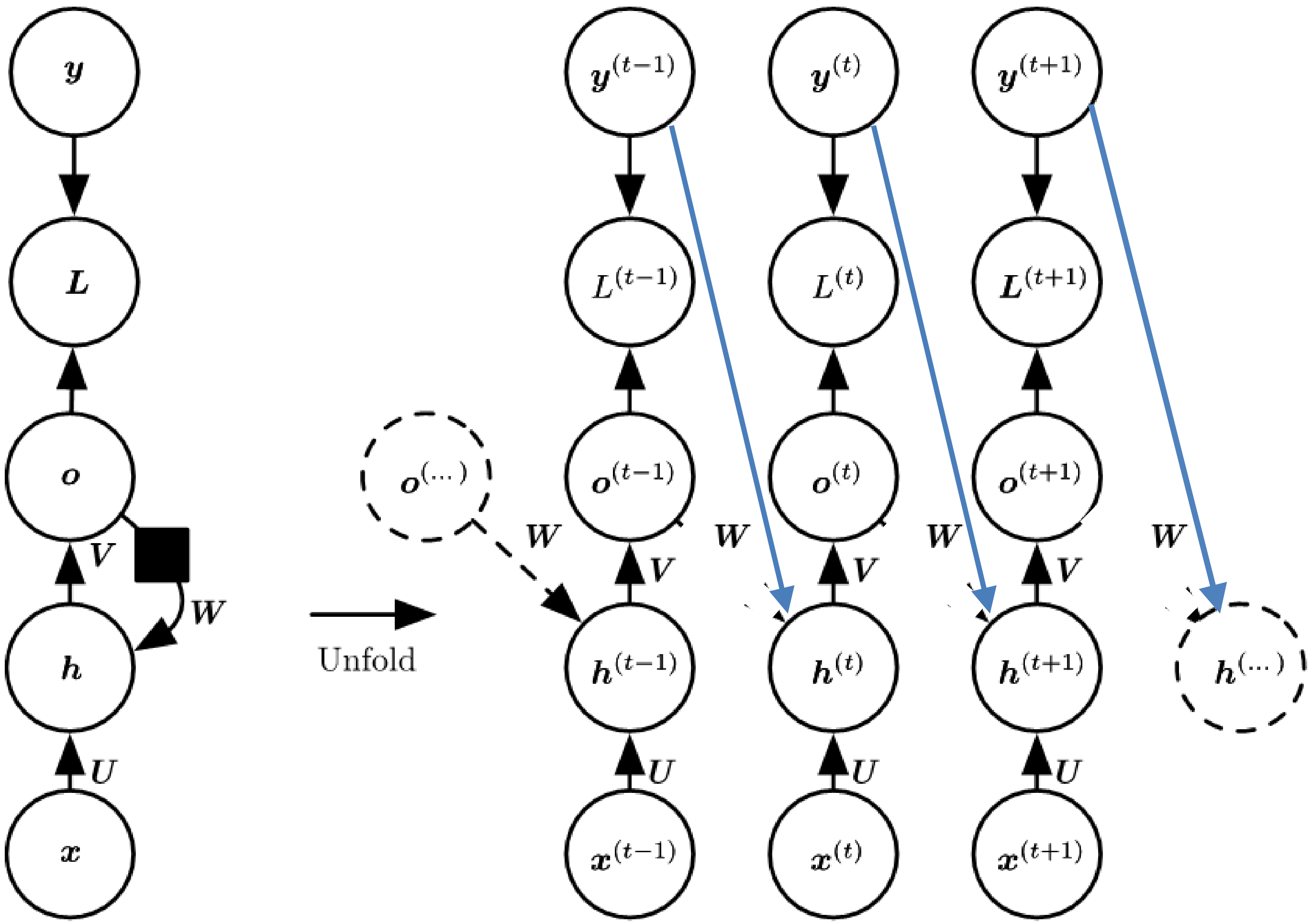
- Organize variables according to time with single update rule
- Finite set of relationships may extend to infinite sequences
- *h* acts as “memory state” summarizing relevant history



[Goodfellow, Bengio, Courville 2016]

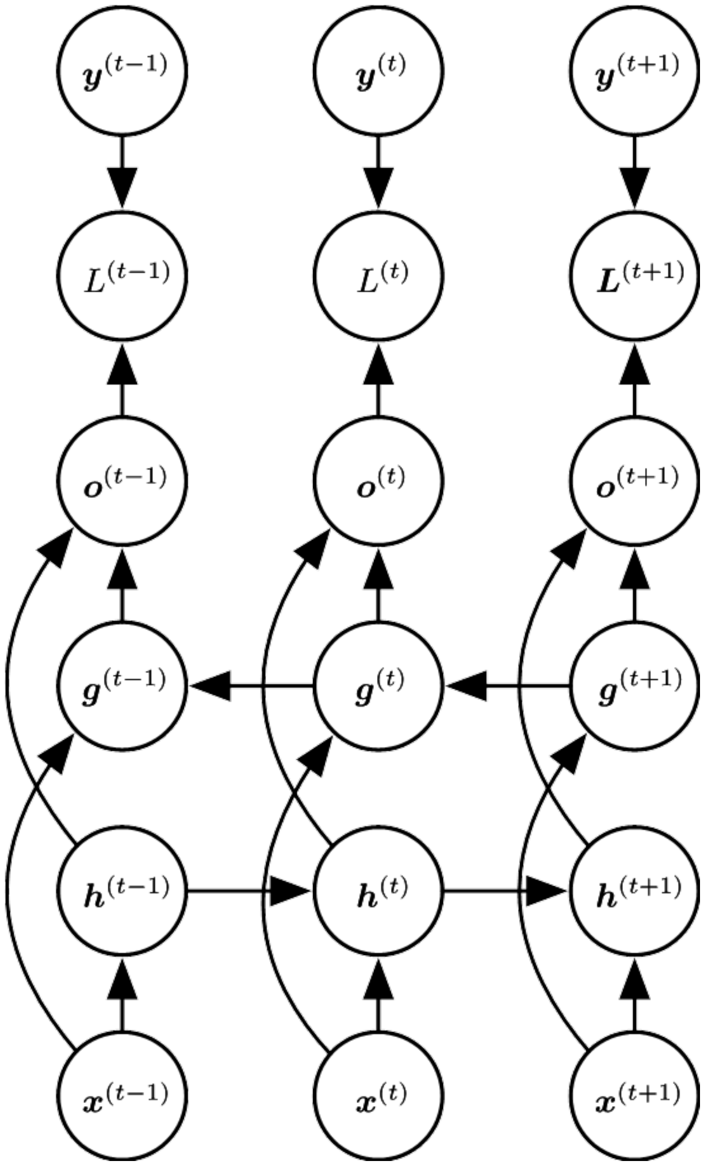
Recurrence only through output

- Avoid backprop through time
- Mitigation: Teacher forcing
  - Use actual or expected output from the training dataset at current time y(t) as input o(t) to the next time step, rather than generated output
  - Backprop stops when it reaches y(t-1) via o(t-1)



Bidirectional RNN

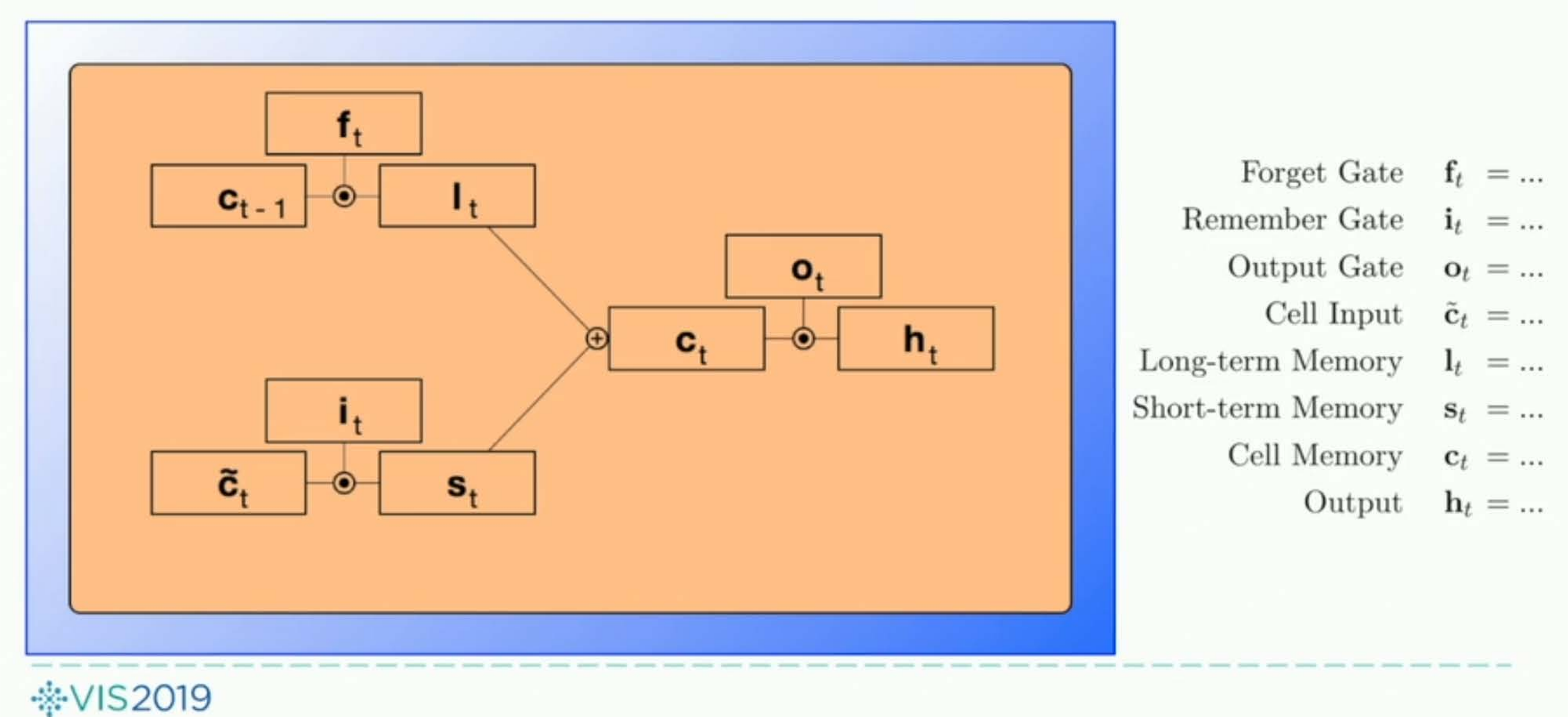
- Later information may be used to reassess previous observations



[Goodfellow, Bengio, Courville 2016]

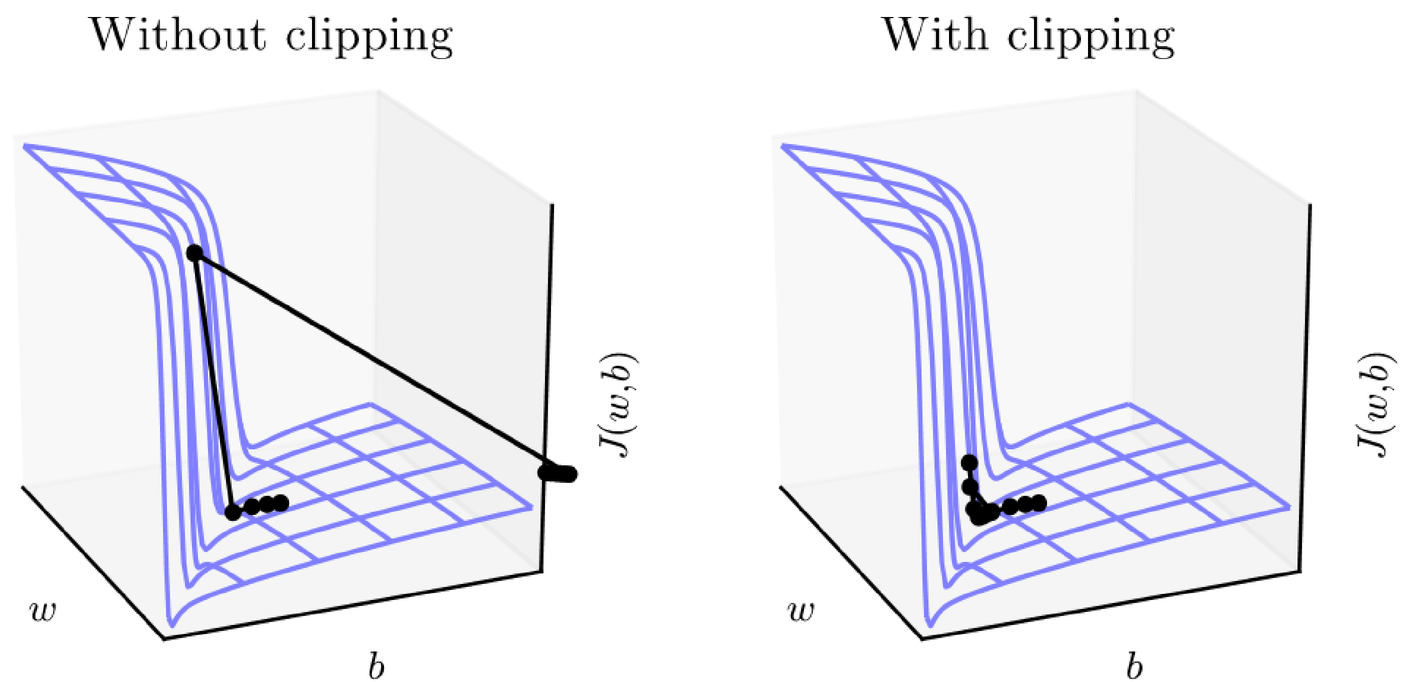
LSTMs

- Use addition over time instead of multiplication



Gradient Clipping

- Add learning rate time gradient to update parameters
- Believe direction of gradient, but not its magnitude



[Goodfellow, Bengio, Courville 2016]

Sentiment Analysis
Word embeddings

Sentiment Analysis

- Computational study of opinions, sentiments, subjectivity, evaluations, attitudes, appraisal, affects, views, emotions, etc., expressed in text

  [B. Liu 2011]

- Aka Opinion mining

Step A: Text processing

- Break up text into sentences
- Tokenize words
- Remove stop words [I, had, the, a, as, there]
- What other preprocessing could be useful?

B1: Words → hash indices

- Each word is a string
- Hash each string to a number

Problem:

- Large vocab leads to large vectors → store as sparse vec or dictionary of counts

B2: Doc → word count vector

- Term frequency (TF)
  - Count the number of occurrences of each string in each doc
- Frequent words with less meaning dominate
- Scale down with a measure of ubiquity
  - inverse doc frequency (IDF)
- Semantically equivalent words are **not** grouped together

Better: Use Word2Vec

Distributional Hypothesis

- Words that are used and occur in same context tend to support the same meaning
- “Judge a word by the company it keeps.”
- Dense word representation (word2vec, see Spark ML)
- Word semantics are taken into account

C: Document → average vectors

- Word vectors → clusters, docs → avg cluster vectors
- Use k-means, cluster groups synonyms or topics

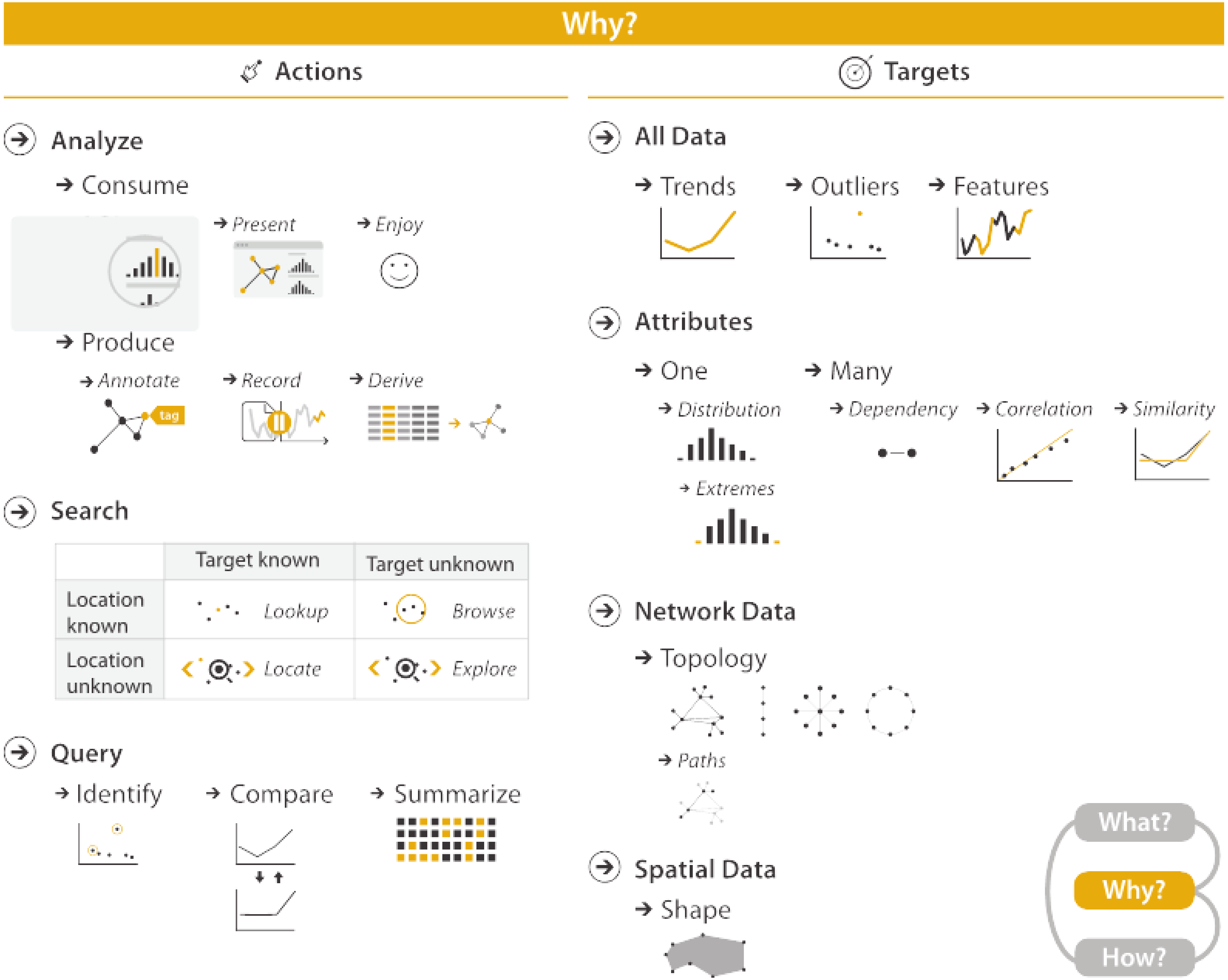
D: Regression / Classification

- Linear regression: star rating
- Logistic regression: likes, smiley types, etc.

Sentiment using LSTMs

- Stanford Sentiment Treebank
  - **https://nlp.stanford.edu/sentiment/treebank.html**
- Sentiment via LSTM using word2vec
  - https://github.com/git-steb/pytorch-sentiment-classification
    - fork of: https://github.com/clairett/pytorch-sentiment-classification/

Visualization Recap:
Data, Task, and Encoding



Tasks

- Actions
  - Analyze
  - Search
  - Query
- Targets
  - Item & Attributes
  - Topology & Shape
  - Models

[T. Munzner, 2014]

Visualization for DL

- **Tensorboard: Visualizing Learning**
- How to use t-SNE efficiently

Model visualization

- **LSTM-Vis**: http://lstm.seas.harvard.edu/client/index.html
- Building blocks of interpretability

Sources

- I. Goodfellow, Y. Bengio, A. Courville "Deep Learning" MIT Press 2016 [link]
- Apala Guha's CMPT 733 slides